

IN THE CLAIMS:

The following are currently pending claims and proposed amendments to claims 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16, 18, 19, 20, 21. Claims 22 through 28 have been added. Claim 1 has been cancelled without prejudice. Claim 7 has been amended to place it in independent form. Claim 9 has been cancelled without prejudice. Claim 15 has been amended to place it in independent form. Claim 17 has been cancelled without prejudice. Claim 21 has been amended to place it independent form. (All claims listed)

1. (Cancelled) ~~A method of responding to a URL request;~~
~~receiving the URL request from a client computer;~~
~~determining a cache in a cache array that stores a URL corresponding to the URL~~
~~request; and~~
~~redirecting the URL request to the determined cache.~~

2. (Currently Amended) The method of claim 1 7, wherein the cache array comprises a plurality of caches and provides for redirection among the plurality of caches.

3. (Currently Amended) The method of claim 1 7, wherein the cache array is a Cache Array Routing Protocol based array.

4. (Currently Amended) The method of claim 1 7, wherein the cache is determined by executing a deterministic algorithm.

DubB1
5. (As Filed) The method of claim 4, wherein the deterministic algorithm is a Cache Array Routing Protocol algorithm.

6. (Currently Amended) The method of claim 4, wherein the URL request is received from an Internet web browser executed on said client computer.

7. (Currently Amended) A method of responding to a URL request comprising:
receiving the URL request from a client computer;
determining a cache in a cache array that stores a URL corresponding to the URL request; and

redirecting the URL request to the determined cache;

A
The method of claim 1, wherein said determining and redirecting is executed by a network processor that is transparent to said client computer.

8. (Currently Amended) The method of claim 7, wherein the network processor is not part of separate from the cache array.

9. (Cancelled) ~~A computer readable medium having instructions stored thereon that when executed by a processor cause the processor, after receiving a URL request from a client computer, to:~~

~~determine a cache in a cache array that stores a URL corresponding to the URL request; and~~

~~redirect the URL request to the determined cache.~~

10. (Currently Amended) The computer readable medium of claim 9 15, wherein the cache array comprises a plurality of caches and provides for redirection among the plurality of caches.

11. (Currently Amended) The computer readable medium of claim 9 15, wherein the cache array is a Cache Array Routing Protocol based array.

12. (Currently Amended) The computer readable medium of claim 9 15, wherein the cache is determined by executing a deterministic algorithm.

13. (As Filed) The computer readable medium of claim 12, wherein the deterministic algorithm is a Cache Array Routing Protocol algorithm.

14. (Currently Amended) The computer readable medium of claim 9 15, wherein the URL request is received from an Internet web browser executed on said client computer.

15. (Currently Amended) A computer readable medium having instructions stored thereon that when executed by a processor cause the processor, after receiving a URL request from a client computer, to:
determine a cache in a cache array that stores a URL corresponding to the URL request; and

Sub B1
redirect the URL request to the determined cache;

The computer readable medium of claim 9, wherein said determine and redirect operations are executed by a network processor that is transparent to said client computer.

16. (Currently Amended) The computer readable medium of claim 15, wherein the network processor is ~~not part of~~ separate from the cache array.

AX
17. (Cancelled) ~~A communication network comprising:~~

~~a network processor; and~~

~~a cache array coupled to said network processor;~~

~~wherein said network processor is programmed to receive a URL request from a client computer, determine a cache in the cache array that stores a URL corresponding to the URL request, and redirect the URL request to the determined cache.~~

Sub B1
18. (Currently Amended) The communication network of claim 17 21, wherein said network processor is ~~not part of~~ separate from said cache array.

19. (Currently Amended) The communication network of claim 17 21, wherein said cache array comprises a plurality of cache servers.

20. (Currently Amended) The communication network of claim 17 21, wherein said cache array is a Cache Array Routing Protocol based array.

Sub B1

21. (Currently Amended) A communication network comprising:

a network processor; and

a cache array coupled to said network processor;

wherein said network processor is programmed to receive a URL request from a client computer, determine a cache in the cache array that stores a URL corresponding to the URL request, and redirect the URL request to the determined cache and

The communication network of claim 17, wherein said network processor is transparent to the said client computer.

22. (Added) A method of routing a URL request comprising:

intercepting, by a network processor, a URL request from a client computer directed to a cache array;

determining a cache in a cache array that stores a URL corresponding to said URL request; and

transmitting said URL request directly into the determined cache.

23. (Added) The method of claim 22, wherein the cache array comprises a plurality of caches and provides for redirection among the plurality of caches.

24. (Added) The method of claim 22, wherein the cache array is a Cache Array Routing Protocol based array.

25. (Added) The method of claim 22, wherein said determining operation includes executing a deterministic algorithm.

26. (Added) The method of claim 22, wherein the deterministic algorithm is a Cache Array Routing Protocol algorithm.

27. (Added) The method of claim 22, wherein the URL request is received from said client computer executing an Internet web browser.

28. (Added) The method of claim 22, wherein the network processor is separate from the cache array.
